mod_glob

Quick Start

A common lament about HTTP servers is the inability to use globbing to retrieve multiple files (as you may be familiar with from FTP servers). This Apache httpd module provides this capability!

You can download the module by cloning the repository using the following command:

```
> git clone https://git.earthdata.nasa.gov/scm/hdd/mod_glob.git mod_glob
```

or you may browse the repository on-line at:

https://git.earthdata.nasa.gov/projects/HDD/repos/mod_glob/browse

Introduction

mod_glob is a drop-in module for Apache httpd version 2.2 and later, that can be used to provide globbing type file downloads. It is configured using the standard Apache configuration directives, and provides two main features. The first is to provide a listing of all the files matching a given glob pattern, including the total file count and data volume. The second is the ability to download all the files matching a given glob pattern as an archive.

For example, a user may request multiple files from a globbing enabled server as follows:

```
> curl "http://myserver.com/data/file*.hdf" -o files.tar
```

If the user wishes to see what files will be returned before they download the archive, a listing can be requested first:

```
> curl "http://myserver.com/data/file*.hdf?list"
```

The standard globbing wildcards are supported:

***** Any zero or more characters

? Any single character. Note that '?' is used in URLs to determine the start of query parameters, and therefore cannot be used directly. It can be URL encoded as %3F, or a @ may be used instead.

[] Any character from a set or range of characters. An ! may be used to invert the set (i.e. any character except)

Requirements

Please note the following requirements.

- mod_glob has only been tested against Apache versions 2.2.22, 2.2.27, and 2.2.29, but should work with most 2.2 and any later versions.
- *apxs* should be available. This is a command line tool used to build and install Apache modules.

Installation

The mod_glob source code is available from the ECC stash repository. It can be retrieved using git:

```
> git clone https://git.earthdata.nasa.gov/scm/hdd/mod_glob.git mod_glob
```

This will create a subdirectory as follows:

```
mod_glob --- README
         |-- LICENSE
         |-- mod_glob.c
         |-- mod_glob.h
         |-- mod_glob_cfg.c
         |-- mod_glob_tar.c
```

To build and install the module, go into the mod_glob source directory and use *apxs*:

```
> cd mod_glob
> apxs -i -n mod_glob -c mod_glob.c mod_glob_cfg.c mod_glob_tar.c
```

In order for the new module to be loaded and used by Apache, you will need to restart httpd. However, it is a good idea to configure the module first.

Configuration

Configuration of Apache is generally done through httpd.conf, although there is a growing trend of different OS distributions to change that, so you may need to do a bit of searching to find the correct configuration file. The first step in configuring mod_glob is to load it into Apache. This is done using the LoadModule directive (apxs may add this directive for you).

```
LoadModule glob_module    modules/mod_glob.so
```

This directive should go at the server root level. Once the module is loaded, you can use configuration directives specific to mod_glob to configure the module.

Apache configuration directives generally fall into two main categories, *server level directives*, and *directory level directives*.

Server level directives are configuration directives that are put at the server or virtual host level (i.e. not inside a directory/location/files configuration block).

Directory level directives are those that go inside a directory/location/files configuration block.

mod_glob only has directory level configuration directives. These directives are:

GlobEnable Enables file globbing for the specified directory or location. By default, globbing is disabled for all directories/locations, and must be explicitly enabled. Valid values are `true/false/yes/no/1/0`.

GlobArchive Specifies the type of archive to use when not explicitly requested by the user. By default, `tar` archives are used (note,

currently mod_glob only supports tar archives. zip support will be coming soon).

GlobMaxSize Specifies the maximum amount of data, in bytes, that a user may request in a single archive. Suffixes such as 'Kb', 'Mb', and 'Gb' (or just 'K', 'M', 'G') may be used, but the value must be an integer. Note that this limit does not apply when requesting a listing, but the listing will contain a warning that the limit has been exceeded. By default, the total data volume is unlimited.

GlobMaxFiles Specifies the maximum number of files that a user may request in a single archive. Note that this limit does not apply when requesting a listing, but the listing will contain a warning that the limit has been exceeded. By default, the maximum number of files is unlimited.

GlobTarUserName Specifies the user name that will be placed into tar archives. This is a string value up to 31 characters in length, but should not contain spaces. This defaults to 'nobody'.

GlobTarGroupName Specifies the group name that will be placed into tar archives. This is a string value up to 31 characters in length, but should not contain spaces. This defaults to 'nobody'.

GlobUseSubRequests Specifies that directory listings should use sub-request to check whether or not a user has access to a file. If your file server users any form of authentication to restrict access to files in a directory for which globbing is enabled, then this option should be enabled, otherwise users will be able to access the files using a glob. By default, this option is disabled, since it incurs a performance penalty. Valid values are `true/false/yes/no/1/0`.

All configuration directives are inherited - any values not explicitly given in a sub-directory will be inherited from the parent directory (or defaulted if no provided in the parent directory).

Example Configurations

Example 1

A simple configuration enabling globbing on a data directory. It places a 1000 maximum file limit, with up to 2Gb of data in a single archive.

```
<Directory "/data/online_archive/fs1/data">

    Options +Indexes

    GlobEnable          true
    GlobArchive         tar
    GlobMaxSize         2Gb
    GlobMaxFiles        1000

</Directory>
```